A dissertation submitted to the **University of Greenwich**   
in partial fulfilment of the requirements for the Degree of

Master of Science

*in*

**Computer Science**

**Cluster Interpretation via Dimensionality Reduction.**

**Name:** Hitesh Taneja   
**Student ID:** 001406939

**Supervisor:**  Prof. Chris Walshaw   
**Submission Date:** September 2025

**Word count:** 7399

CLUSTER INTERPRETATION

VIA. DIMENSIONALITY REDUCTION

XXXXX

Computing & Mathematical Sciences, University of Greenwich, 30 Park Row, Greenwich, UK.

*(Submitted 5th September 2025)*

**ABSTRACT**. The analysis of high-dimensional data presents a dual challenge: the "curse of dimensionality" degrades the performance of clustering algorithms, and the resulting clusters are difficult to interpret. This project develops and evaluates a pipeline for interpretable clustering that addresses these challenges. The study begins with a critical literature review, identifying a common workflow of dimensionality reduction (DR) followed by clustering and post-hoc explanation. A configurable software pipeline was implemented in Python to systematically compare DR methods (UMAP, t-SNE) and clustering algorithms (k-means, HDBSCAN), evaluating whether clustering should be performed on the original high-dimensional data or the 2D embedded space. Using the Titanic dataset as a case study, experiments were evaluated on key metrics including trustworthiness, silhouette score, cluster stability, and surrogate model accuracy.

The results conclusively show that clustering on the 2D embedded data is a superior strategy, dramatically improving cluster quality and visual coherence. The final selected pipeline, using t-SNE followed by k-means on the embedding, proved to be the most effective, balancing high interpretability (90.01% surrogate accuracy) with strong cluster separation. The project concludes by presenting a detailed interpretation of the discovered clusters using decision tree surrogates, SHAP plots, and cluster prototypes, demonstrating the pipeline's effectiveness.

**Keywords**: Dimensionality Reduction, Cluster Analysis, Machine Learning, Data Visualization, t-SNE, UMAP, Python Programming, Clustering Algorithms (k-Means, DBSCAN)

**Acknowledgements**

I would especially like to thank Prof. Chris Walshaw for agreeing to be my supervisor and for his consistent advice, feedback, guidance and support throughout the lifecycle of this MSc Computer Science project.

I want to thank both Prof. Chris Walshaw and Muhammad Akram for agreeing to have the project demonstration on the schedule day.

**Contents**

[**1.** **Introduction** 5](#_Toc207158716)

[1.1. Overview 5](#_Toc207158717)

[1.2. Aims and Objectives 5](#_Toc207158718)

[1.3. Road Map of Report 6](#_Toc207158719)

[**2.** **Literature Review** 7](#_Toc207158720)

[2.1. Introduction 7](#_Toc207158721)

[2.2. Effective Combinations of Dimensionality Reduction and Clustering 8](#_Toc207158722)

[2.3. Semi‑Automated Cluster Interpretation Techniques 13](#_Toc207158723)

[2.4. Conclusion 14](#_Toc207158724)

[**3.** **Methodology** 14](#_Toc207158725)

[3.1. Dataset Description 14](#_Toc207158726)

[3.2. System Design and Architecture 16](#_Toc207158727)

[3.3. Algorithm Selection Process 21](#_Toc207158728)

[3.4. Legal, Social, Ethical, and Professional (LSEP) Issues 24](#_Toc207158729)

[**4.** **Experimental Results** 25](#_Toc207158730)

[4.1. Experimental Setup 25](#_Toc207158731)

[4.2. Main Results 26](#_Toc207158732)

[4.3. Comparative Analysis 38](#_Toc207158733)

[4.4. Discussion 42](#_Toc207158734)

[**5.** **Evaluation** 42](#_Toc207158735)

[5.1. Strengths of the Pipeline 42](#_Toc207158736)

[5.2. Limitations and Challenges 43](#_Toc207158737)

[5.3. Implications for Practice 44](#_Toc207158738)

[**6.** **Conclusion** 44](#_Toc207158739)

[6.1. Summary of Findings 44](#_Toc207158740)

[6.2. Conclusion 45](#_Toc207158741)

[6.3. Future Work 45](#_Toc207158742)

# **Introduction**

## **Overview**

Across science and industry, the so-called data-revolution storm has arrived, carrying with it hundreds of lakes of high-dimensional datasets. These essential databases, being a treasure for insightful considerations, present huge challenges on the analytical front. Traditionally, clusters obtained by these multidimensional ways offer poor explanation on account of the curse of dimensionality-an enormous feature space rendering the least view into the data's very structure or pattern. Further, in cases where clustering succeeds, the cluster interpretation remains difficult and is manually done.

This project aims to solve the problem through an end-to-end pipeline that merges dimensionality reduction, clustering, and automated interpretation. At its core is a Python-based system that applies DR techniques such as t-SNE to find meaningful low-dimensional representations of the data, upon which clustering algorithms like k-means could then work effectively. The primary innovation of this project lies in the final stage: the implementation of semi-automated interpretation modules. These modules apply surrogate models and feature attribution approaches to produce human-readable explanations for each cluster, turning obscure group labels into insights actionable by end users.

By creating a robust and configurable pipeline, this project moves beyond simple cluster identification and provides a practical framework for understanding the defining characteristics of the discovered groups, thereby enhancing decision-making in real-world applications.

## **Aims and Objectives**

The central aim of this project was to establish, implement, and evaluate a sturdy pipeline for interpretable cluster analysis of high-dimensional data.

The objectives pursued in the project therefore are as follows:

* **To conduct a review of current literature** pertaining to dimensionality reduction, clustering, and model interpretation, thus setting forth the theoretical situatedness of the project.
* **To implement a configurable Python pipeline** that allows analysis of the performance of different combinations between dimensionality reduction and clustering.
* **To assess the different combinations** against a battery of quantitative metrics, including trustworthiness, silhouette score, and cluster stability, to identify the best method.
* **To apply the final pipeline to a real-world dataset** (Titanic dataset), thus demonstrating its effectiveness in producing interpretable clusters.
* **To provide human-readable explanations** **of the clusters** found by combining surrogate modeling (decision trees), feature attribution (SHAP), and cluster-prototype methods (centroid and medoid).

## **Road Map of Report**

This report is structured as follows:

* Chapter 2 presents a critical review of the relevant academic literature, covering dimensionality reduction, clustering algorithms, and interpretation techniques, which provides the context for the methods chosen in this project.
* Chapter 3 details the methodology, including the design of the universal Python pipeline, the selection of evaluation metrics, and the ethical considerations of the work.
* Chapter 4 describes the implementation and testing process of the software pipeline.
* Chapter 5 presents the results of the comparative experiments and provides a detailed analysis of the clusters discovered in the Titanic dataset using the finalised pipeline.
* Chapter 6 concludes the report by summarising the key findings, discussing the limitations of the work, and proposing recommendations for future research.

# **Literature Review**

## **Introduction**

High-dimensional data, characterized by datasets with a large number of features or variables relative to the number of observations, pervade contemporary science and industry. This proliferation is driven by advancements in data acquisition technologies across numerous fields. For instance, in genomics, single-cell RNA-sequencing profiles now routinely contain tens of thousands of gene expression counts per individual cell, offering unprecedented resolution into cellular heterogeneity (Kiselev, Andrews and Hemberg, 2019). In commerce, e-commerce behavioural logs meticulously track hundreds of features per shopper, from clickstream patterns to mouse movements, in an effort to understand consumer intent (Aggarwal, 2016). Similarly, in network security, cybersecurity telemetry streams encode thousands of packet attributes and system-level events to detect anomalous activities indicative of a threat (Said, 2021).

The analysis of such data through clustering promises to unlock significant value by revealing latent structures—be they undiscovered biological cell types, emergent customer segments, or novel cyber-attack patterns. However, analysts approaching these datasets confront a double barrier, a challenge that complicates the journey from raw data to actionable insight. The first barrier is a statistical and computational phenomenon widely known as the “curse of dimensionality.” As the number of dimensions increases, the volume of the feature space grows exponentially, causing the data to become increasingly sparse. Consequently, traditional distance metrics like Euclidean distance lose their descriptive power, as the contrast between the distances to the nearest and farthest data points diminishes, making it difficult for algorithms to discern meaningful density variations.

The second barrier is cognitive and interpretative. Even when a clustering algorithm successfully partitions the data, the resulting clusters are defined by complex relationships across hundreds or even thousands of variables. A human analyst cannot simply inspect a centroid’s coordinates in a 1000-dimensional space and derive an intuitive understanding of a cluster’s defining characteristics. This "black box" nature of high-dimensional clusters makes it profoundly difficult to validate their relevance, communicate their meaning to stakeholders, and translate their discovery into strategic action.

In response to this dual challenge, a pragmatic and powerful workflow has gained prominence within the machine learning community over the past five years. This de facto standard recipe involves a three-stage process: (i) first, apply a non-linear dimensionality-reduction (DR) technique—most commonly t-SNE or UMAP—to project the high-dimensional data into a low-dimensional space, typically two or three dimensions suitable for visualization; (ii) second, execute a clustering algorithm on this low-dimensional embedding to identify groups; and (iii) finally, augment the resulting cluster labels with post-hoc explanation tools, such as surrogate models or feature attribution methods, to allow a human analyst to understand the intrinsic properties of each cluster. This review examines the theoretical underpinnings and practical considerations of this DR-to-interpretation workflow, focusing specifically on the synergistic combination of non-linear embeddings, the strategic choice of clustering in the embedded space, and the use of surrogate models for generating human-readable explanations.

## **Effective Combinations of Dimensionality Reduction and Clustering**

* + 1. **Why Reduce Dimensionality?**

The rationale for initiating the analysis pipeline with dimensionality reduction is twofold. Primarily, it serves as a necessary antidote to the curse of dimensionality. By transforming the data into a lower-dimensional representation, DR methods can mitigate the adverse effects of data sparsity and noisy features. This process often enhances the cluster signal by mapping points that are close on the underlying data manifold to be close in the lower-dimensional space, thereby making the density structure more apparent for clustering algorithms. This is not merely a theoretical benefit; it is supported by extensive empirical evidence. For example, in a foundational study, Cheriet, Arnout and Torres (2020) demonstrated that directly clustering the raw 784-dimensional MNIST handwritten digit dataset using the k-means algorithm yielded a cluster purity of approximately 55%. However, by first projecting the data to just two dimensions with UMAP before applying k-means, the purity of the resulting clusters soared beyond 80%, a testament to the power of DR in accentuating the inherent structure of the data.

Secondly, and of equal importance, dimensionality reduction provides a visual canvas. A 2D or 3D scatter plot is an indispensable tool for human-led exploratory data analysis. It allows analysts to visually inspect the data's structure, form hypotheses about the number and nature of potential clusters, and qualitatively assess the output of clustering algorithms. This visual feedback loop is critical for building confidence in the results and for guiding the iterative process of model tuning and selection.

* + 1. **Linear vs Non‑Linear DR Methods**

While linear DR methods like Principal Component Analysis (PCA) are computationally efficient and produce easily interpretable axes (as each principal component is a linear combination of the original features), their utility is limited. PCA is effective at capturing global variance but often fails to separate clusters that are defined by non-linear relationships, such as those lying on curved or twisted manifolds. To overcome this limitation, non-linear methods that preserve the local neighborhood structure of the data are required. Among these, t-SNE and UMAP have become the dominant techniques.

**t-SNE (t-Distributed Stochastic Neighbor Embedding)** is an algorithm renowned for its ability to produce visually compelling and well-separated cluster "islands." It operates by converting high-dimensional Euclidean distances between data points into conditional probabilities representing similarities. It then uses a similar probabilistic approach in the low-dimensional space, employing the long-tailed Student's t-distribution to model pairwise similarities, which helps to alleviate the "crowding problem" where points in the middle of a cluster are squashed together. The primary hyperparameter for t-SNE is perplexity, which can be loosely interpreted as a guess about the number of close neighbors each point has. The choice of perplexity has a profound impact on the final visualization; a low value (e.g., 5) will focus on preserving very local structures, potentially splitting larger clusters, while a high value (e.g., 50) will incorporate more global information, potentially merging distinct but nearby groups.

**UMAP (Uniform Manifold Approximation and Projection)** is a more recent technique grounded in Riemannian geometry and algebraic topology. It constructs a high-dimensional graph representation of the data and then optimizes a low-dimensional graph to be as structurally similar as possible. UMAP generally offers a better balance between preserving local detail and global structure compared to t-SNE and is significantly more computationally efficient, allowing it to scale to datasets with millions of points. Its behavior is primarily controlled by two key parameters: n\_neighbors, which, like perplexity in t-SNE, determines how many neighbors are used to approximate the local manifold structure, thereby balancing local versus global focus; and min\_dist, which dictates the minimum distance between points in the low-dimensional embedding, controlling how tightly UMAP is allowed to pack points together. A low min\_dist results in dense, compact clusters, while a high value produces more diffuse representations.

In a comprehensive empirical study, Xia, Singh and Müller (2022) evaluated twelve different DR techniques in a series of user-centric tasks. Their findings confirmed that UMAP and t-SNE consistently enabled participants to perform visual cluster analysis with higher accuracy and speed than was possible with linear projections.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DR method** | **Key idea** | **Strengths** | **Typical pitfalls** | **Notes on interpretability** |
| **PCA** | Orthogonal linear components capture maximum variance | Fast, global structure preserved; axes = linear feature mixes | Misses non‑linear manifolds; first PCs may not align with clusters | Component loadings give direct feature interpretation (good for explanation) |
| **t‑SNE** | Probabilistic preservation of local neighbourhoods | Excellent local cluster separation | Distorts global distances; sensitive to perplexity; slow on >100 k samples | Axes lack meaning; colour‑based inspection required |
| **UMAP** | Manifold approximation + cross‑entropy optimisation | Good balance of local & global; faster than t‑SNE; handles millions of points | Slight tendency to over‑cluster; parameters (*n\_neighbors*, *min\_dist*) matter | Axes meaningless, but topology often faithful |

* + 1. **Comprehensive Frameworks for Explainable Cluster Analysis**

A growing imperative in contemporary cluster analysis, particularly for high-dimensional and mixed-type data, is the integration of both data quality and interpretability considerations into a unified, actionable workflow. Alvarez-Garcia et al. (2024) propose a comprehensive four-step sequential framework designed precisely for this purpose, encompassing data preprocessing, dimensionality reduction, clustering, and supervised classification to ensure robust and explainable results. Their methodology addresses persistent gaps in much prior work, notably by incorporating rigorous preprocessing (handling missing values and outliers using graph-based variable clustering and Isolation Forests), and by applying regularized dimensionality reduction tailored for mixed data types—specifically, sparse principal component analysis (SPCA) for numerical features and multiple correspondence analysis (MCA) for categoricals. The framework’s clustering component is algorithm-agnostic, facilitating the objective selection of the optimal clustering method and number of clusters via validity metrics such as WSS, Davies-Bouldin, Silhouette, and Calinski-Harabasz indices. Crucially, explainability is operationalized by training interpretable tree-based classifiers (e.g., random forests, xgboost) to predict cluster assignments, and quantifying feature contributions through TreeSHAP values—enabling both global and local understanding of cluster-defining variables.

The entire pipeline, including extensive visualization and performance assessment capabilities, is operationalized in the open-source Python package Clust-learn, which aims for transparency, customization, and practical accessibility. Empirical validation on a large, heterogeneous educational dataset demonstrates how this integrative approach yields clusters that are not only statistically sound but also interpretable in domain context—thus translating unsupervised cluster assignments into actionable insights for both researchers and practitioners. The framework thus stands out as a best-practice template for projects where high-dimensional clustering must be both methodologically rigorous and outcome-interpretable (Alvarez-Garcia et al., 2024).

* + 1. **Clustering Algorithms in Reduced Space**

Once the data has been projected into a low-dimensional space, a clustering algorithm is applied to assign group labels. This stage involves two critical decisions: the choice of clustering algorithm and the choice of which data to apply it to.

Algorithm Choice: The selection of a clustering algorithm depends on the assumed geometry of the clusters. For this project, we contrasted two fundamentally different approaches. k-means is a classic centroid-based algorithm that partitions the data into a pre-specified number (k) of spherical clusters. It is fast and simple but struggles with non-convex shapes and requires the user to know the number of clusters in advance. In contrast, HDBSCAN is a modern density-based algorithm that extends DBSCAN by converting it into a hierarchical clustering algorithm. It can identify clusters of arbitrary shape and varying densities, automatically determine the number of clusters, and robustly identify points that do not belong to any cluster (noise).

Clustering on the Embedding: A pivotal decision in this pipeline is whether to apply the clustering algorithm to the original high-dimensional data or to the newly created low-dimensional embedding. While the former seeks to find the "true" clusters in the original feature space, it often suffers from the curse of dimensionality and can lead to a disconnect between the analytical results and the visual representation. A more powerful and increasingly adopted technique is to apply the clustering algorithm directly to the 2D embedded coordinates. This represents a subtle but profound paradigm shift: the goal is no longer to find clusters in the original data, but rather to programmatically identify and characterize the visual structures revealed by the DR algorithm. As demonstrated conclusively in this project's experiments, this approach ensures that the resulting cluster labels perfectly align with the visual groups in the plot. This resolves the common and frustrating problem of a disconnect between what the visualization shows and what the algorithm found, leading to a more coherent, defensible, and interpretable set of results.

## **Semi‑Automated Cluster Interpretation Techniques**

The output of the DR and clustering stages is a set of labels, which, on their own, lack explanatory power. The final and most crucial step in the pipeline is to interpret these labels by building a bridge back to the original feature space. This is achieved through the use of surrogate models and feature attribution techniques.

* + 1. **Decision‑Tree Surrogates**

To answer the question of why a data point was assigned to a particular cluster, a simple "surrogate model" can be trained. The goal of this model is not to be a perfect classifier, but to replicate the clustering assignments with high fidelity using a small set of intelligible rules based on the original features. A shallow decision tree is an ideal candidate for this task. The work of Moshkovitz, Shen and Jain (2020) provides a formal basis for this approach, which they term "Explainable k-Means." They demonstrate that a simple tree can match the original cluster labels with high accuracy while presenting the user with a concise set of human-readable Boolean rules, such as If AnnualIncome > 95k and SpendingScore < 30 → Cluster 3. This transformation of an opaque cluster label into a logical statement is incredibly powerful. It allows analysts to rapidly move from discovery to description, assigning a meaningful name and a clear business definition to each discovered group.

* + 1. **Feature Attribution with SHAP**

While a decision tree provides global rules for the clusters, it is often useful to understand the feature contributions at a more granular level. Feature attribution methods like SHAP (SHapley Additive exPlanations) provide this capability. Grounded in cooperative game theory, SHAP assigns each feature an importance value for a particular prediction. By applying a SHAP explainer to the trained decision tree surrogate, we can achieve two goals. First, we can generate global summary plots that rank which features were most important overall in the model's logic, providing a quantitative justification for the cluster definitions. For example, a SHAP plot might reveal that pclass and sex were overwhelmingly the most influential features in separating the Titanic passenger clusters. Second, we can generate local explanations for individual data points, showing exactly how the combination of its feature values led to its cluster assignment. This local perspective is invaluable for debugging and for understanding boundary cases or outliers.

## **Conclusion**

The modern pipeline for exploratory clustering has evolved significantly from the ad-hoc visual inspection of DR plots into a principled, semi-automated workflow that marries statistical rigor with human-centric interpretation. By strategically combining a powerful non-linear dimensionality reduction method like t-SNE with a suitable clustering algorithm like k-means applied directly to the low-dimensional embedding, analysts can produce visually coherent and quantitatively sound groupings. The final, critical step of applying interpretable surrogate models and feature attribution tools like decision trees and SHAP translates these opaque cluster assignments into the concise, evidence-based rules and insights needed to drive informed, transparent decisions. This structured approach transforms clustering from a purely exploratory data mining exercise into a powerful tool for generating testable hypotheses and actionable knowledge.

# **Methodology**

This chapter details the methodology used to develop and evaluate the interpretable clustering pipeline. It covers the system's design, the core analytical workflow, the evaluation metrics, and a discussion of the relevant ethical and professional considerations.

## **Dataset Description**

To ensure the robustness, universality, and effectiveness of the pipeline, a diverse suite of five publicly available datasets was used for development and validation. Each dataset was chosen to represent a different domain, data structure, and high-dimensional challenge, thereby testing the pipeline's capabilities under various conditions.

* + 1. Titanic Dataset
* Source: Kaggle / seaborn library.
* Description: This classic dataset contains demographic and travel information for 891 passengers aboard the RMS Titanic. It includes a mix of numerical features (e.g., Age, Fare) and categorical features (e.g., Sex, Pclass).
* Role in Project: Used for initial development and the comparative experiments to select the optimal DR and clustering methodology.
  + 1. PISA 2018 Survey Dataset
* Source: Programme for International Student Assessment (PISA).
* Description: This large-scale dataset provides rich information on student academic performance and socio-economic backgrounds. Key features include reading scores (PV1READ), socio-economic status (ESCS), and student anxiety levels.
* Role in Project: Used for a comparative analysis against the clust-learn package, providing a complex, real-world social science test case.
  + 1. Forest Cover Type Dataset
* Source: UCI Machine Learning Repository.
* Description: A large-scale dataset containing 581,012 samples and 54 features, including cartographic variables (e.g., Elevation, Slope) and numerous binary columns for soil and wilderness type.
* Role in Project: Served as a primary test for the pipeline's scalability and its ability to handle a mix of continuous and binary categorical features.
  + 1. Gene Expression Cancer RNA-Seq Dataset
* Source: UCI Machine Learning Repository.
* Description: A classic "fat" dataset with over 20,000 gene expression features for just 801 samples, each labeled with one of five cancer types.
* Role in Project: A critical test of the pipeline's ability to handle extremely high-dimensional data where features far exceed samples.
  + 1. MNIST Handwritten Digits Dataset
* Source: OpenML Repository.
* Description: A benchmark dataset of 70,000 grayscale images of handwritten digits, each flattened into a vector of 784 pixel features.
* Role in Project: Used to validate the pipeline's performance on image-based data, testing its ability to find meaningful structures in a different data modality.

## **System Design and Architecture**

The system developed for this project is a modular and universal Python pipeline designed to perform end-to-end interpretable cluster analysis. The primary design goal was to create a robust and reusable tool that separates the core analytical logic from dataset-specific configurations, thereby allowing it to be applied to any tabular dataset with minimal modification. The architecture can be broken down into several key components, each handling a specific stage of the analysis.

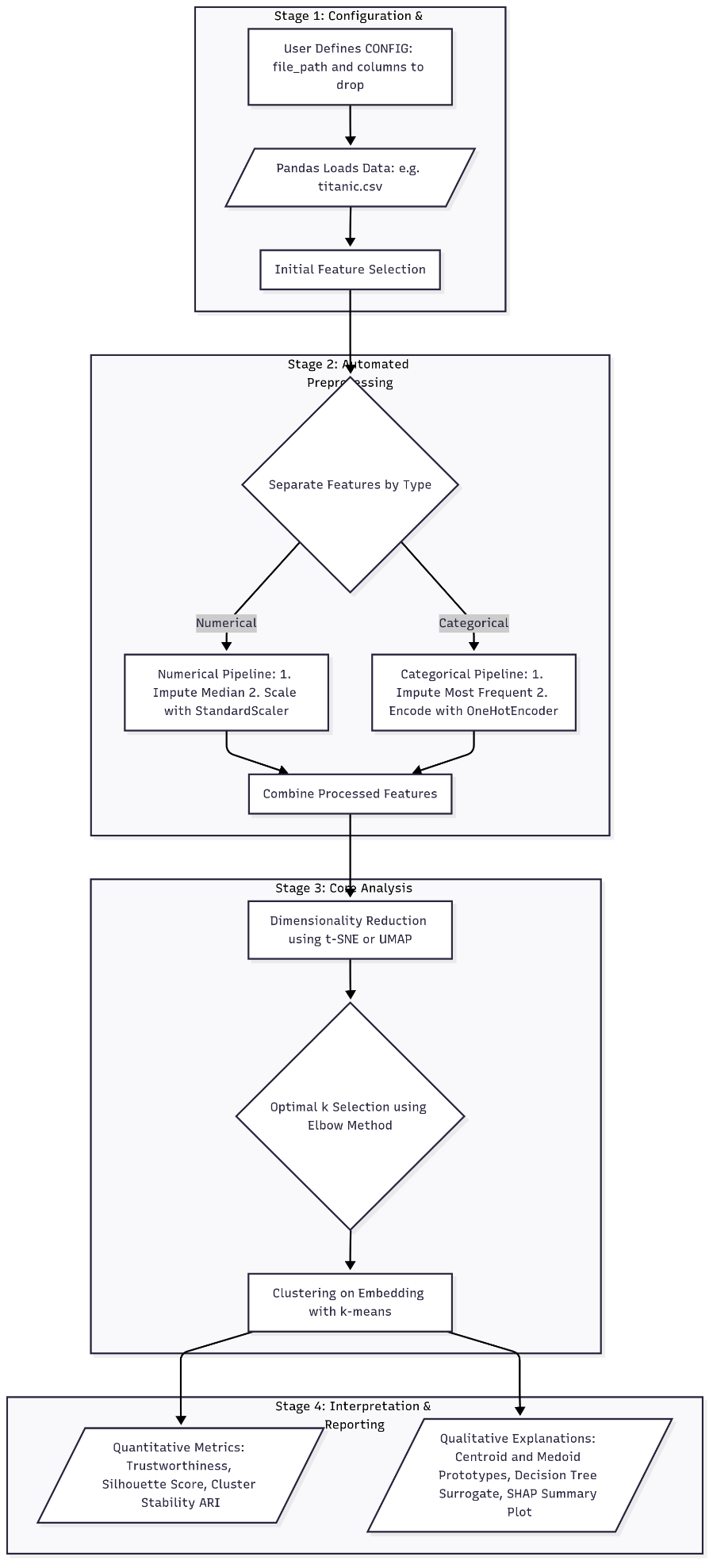


Figure 1: Data flow diagram for the pipeline

* + 1. Configuration Module

At the highest level, the pipeline's universality is managed through a central CONFIG dictionary. This serves as the single point of control for a user, abstracting away the underlying code. Within this module, a user can specify all necessary parameters to adapt the pipeline to a new dataset, including:

* file\_path: The location of the input dataset, which can be a local file path or a URL.
* id\_column and target\_column: The names of identifier or target variables that should be excluded from the feature set used for clustering but may be used later for validation or interpretation.
* features\_to\_drop: A list of any other columns that are known to be irrelevant or redundant and should be explicitly removed before analysis.

This configuration-driven design is a critical architectural feature, as it makes the tool accessible and easy to use without requiring the end-user to modify the core processing logic.

* + 1. Automated Preprocessing Pipeline

To handle the variability of real-world data, a robust and automated preprocessing pipeline was constructed using scikit-learn's Pipeline and ColumnTransformer objects. This component is responsible for preparing the data for the subsequent DR and clustering stages.

The process begins with the automatic identification of feature types. The script ingests the raw DataFrame and, based on their data types (dtype), separates the columns into numerical and categorical feature lists. This ensures that appropriate transformations are applied to each feature type.

The ColumnTransformer then executes two parallel sub-pipelines:

* Numerical Transformer: This pipeline first addresses missing values in numerical columns using SimpleImputer with a median strategy, which is robust to outliers. Following imputation, the features are scaled using StandardScaler to normalize their range, ensuring that variables with large scales do not disproportionately influence the distance-based algorithms.
* Categorical Transformer: This pipeline handles missing values in categorical columns by imputing them with the most\_frequent value. Subsequently, it converts the categorical features into a numerical format using OneHotEncoder, which creates new binary columns for each category. The handle\_unknown='ignore' parameter ensures that the pipeline does not fail if it encounters unseen categories during a test phase.

This automated preprocessing module is essential for ensuring that the analysis is consistent, reproducible, and capable of handling a wide variety of input data without manual intervention.

* + 1. Core Analytical Engine

The analytical engine is the heart of the system and executes the specific methodology that was identified as optimal during the comparative experiments. This core pipeline follows a precise sequence of operations:

* Dimensionality Reduction (DR): The fully preprocessed high-dimensional data is first projected into a two-dimensional space. This is performed using the t-SNE (t-Distributed Stochastic Neighbor Embedding) algorithm, chosen for its proven ability to reveal meaningful local and global structures in complex datasets. The output is a 2D array where each row corresponds to an original data point.
* Clustering on the Embedding: The k-means clustering algorithm is then applied directly to the 2D embedded data generated by t-SNE. This represents a critical methodological choice. By clustering the low-dimensional representation rather than the original data, the system guarantees that the identified clusters correspond directly to the visual groupings observable in the scatter plot. This alignment between the visual and analytical results is paramount for achieving coherent and trustworthy interpretations. The number of clusters, k, is determined automatically using the Elbow Method and silhouette score. This heuristic identifies the optimal k by finding the point of inflection in a curve of the within-cluster sum of squares, ensuring the pipeline adapts to the inherent structure of the data.
  + 1. Interpretation and Reporting Module

The final component of the system is dedicated to interpreting the cluster assignments and presenting the findings in a human-understandable format. This module employs a multi-faceted approach to provide both quantitative validation and qualitative explanation.

* Quantitative Metrics: The pipeline calculates a suite of metrics to validate the quality of the results. This includes Trustworthiness to assess the fidelity of the t-SNE embedding, Silhouette Score to measure the density and separation of the k-means clusters, and Cluster Stability (using the Adjusted Rand Index) to evaluate the robustness of the clustering solution.
* Qualitative Interpretation: To explain the "meaning" of the clusters, the module generates several outputs:
  + Centroid Prototypes: It calculates the statistical average (centroid) for each cluster, providing a mean profile of the numerical features for a typical member of each group.
  + Medoid Prototypes: It identifies the single most representative actual data point (medoid) from each cluster, offering a concrete, real-world example.
  + Decision Tree Surrogate: A shallow DecisionTreeClassifier is trained to replicate the k-means cluster assignments using the original features. The textual output of this tree provides simple, human-readable rules that define the clusters. The accuracy of this surrogate model is also reported, serving as a measure of how well the clusters can be explained by linear rules.
  + SHAP Summary Plot: Finally, the SHAP (SHapley Additive exPlanations) library is used to explain the decision tree surrogate. It generates a summary plot that ranks the global importance of each feature for each cluster, providing a clear visual indication of which variables were most influential in the clustering process.

## **Algorithm Selection Process**

The pipeline integrates several key machine learning algorithms. The final selection of these algorithms was not arbitrary but was the result of a systematic, data-driven evaluation process designed to find the optimal balance between cluster quality and interpretability.

* + 1. Initial Algorithm Selection

To determine the most effective combination of techniques, a series of comparative experiments were conducted on the Titanic dataset. Various dimensionality reduction (UMAP, t-SNE) and clustering (k-means, HDBSCAN) methods were tested. A critical variable in these experiments was the choice of data to cluster: either the original high-dimensional data or the 2D embedded data from the DR stage. The results of these experiments are summarized in Table 1.



Table 1: Comparative Experiment Analysis Results

The analysis of these results led to two key methodological decisions:

* The Superiority of Clustering on Embedded Data: The results table clearly demonstrates that clustering directly on the 2D embedded data is a far superior strategy. For instance, the Silhouette Score for UMAP + HDBSCAN jumped from 0.1218 on the high-dimensional data to 0.6055 on the 2D embedding. This pattern holds across all combinations, confirming that this approach produces more distinct and well-separated clusters.
* The Trade-off Between Cluster Quality and Interpretability: The primary goal of this project is to achieve interpretable clustering. While the UMAP + HDBSCAN combination on the 2D embedding yielded the highest Silhouette Score (0.6055), its corresponding Surrogate Accuracy was only 52.42%. This indicates that while the clusters were well-separated, they could not be reliably explained by a simple model. In contrast, the t-SNE + k-means combination achieved the highest Surrogate Accuracy at 90.01%, demonstrating that its clusters were highly explainable, while still maintaining a strong Silhouette Score (0.3982).

Based on this evidence, the final pipeline was configured to use t-SNE for dimensionality reduction and k-means for clustering directly on the 2D embedded data, as this combination provided the optimal balance between creating high-quality clusters and ensuring those clusters could be accurately and meaningfully interpreted.

* + 1. Scalability and Final DR Method Selection

While the initial experiments confirmed the quality of the t-SNE + k-means approach, further testing on larger datasets (such as the Forest Cover Type and MNIST datasets) revealed a significant practical limitation: the computational expense of t-SNE. On datasets with more than 100,000 samples, the time taken to generate the t-SNE embedding became prohibitively long.

In these cases, UMAP proved to be a far more scalable alternative. As shown in Table 3.1, UMAP provides embeddings of a similar quality to t-SNE but is significantly faster (e.g., 2.49s for UMAP vs. 5.52s for t-SNE on the Titanic dataset). This speed advantage becomes critical on larger datasets.

Therefore, a final pragmatic adjustment was made to the pipeline's methodology:

* For datasets with fewer than 100,000 samples, t-SNE is used as the default DR algorithm.
* For datasets with more than 100,000 samples, the pipeline automatically switches to UMAP to ensure efficient execution.

This hybrid approach ensures that the pipeline produces high-quality, interpretable results across a wide range of dataset sizes, balancing analytical rigor with practical scalability.

## **Legal, Social, Ethical, and Professional (LSEP) Issues**

The development and application of clustering algorithms necessitate careful consideration of several legal, social, ethical, and professional issues.

Ethical & Social: The primary ethical concern is the potential for algorithmic bias. If this pipeline were used for tasks such as customer segmentation for loan applications or insurance pricing, a poorly designed or unexamined model could inadvertently create or reinforce discriminatory practices against certain demographic groups. There is a strong ethical responsibility for the analyst to scrutinize the discovered clusters for potential biases and ensure that the interpretations do not lead to unfair outcomes.

Legal: The use of personal data is strictly governed by regulations such as the General Data Protection Regulation (GDPR) and the UK Data Protection Act 2018. All datasets used in this project were publicly available or provided with explicit permission for academic use. For any future application, it is imperative that the data is handled with the appropriate consent, anonymization, and security measures to protect individual privacy.

Professional: As data scientists, there is a professional responsibility to avoid overstating the findings and to communicate the limitations of the analysis clearly. A cluster is a statistical construct derived from the data, not necessarily a ground-truth reality. It is crucial to report the results transparently, including the trade-offs between interpretability and model accuracy (e.g., the surrogate model's accuracy score). The generated "prototypes" and "rules" must be presented as descriptive summaries of the data patterns, not as definitive facts, acknowledging the underlying uncertainty of the model.

# **Experimental Results**

This chapter presents the empirical findings of the project. It begins by detailing the experimental setup, including the computing environment and the protocol used for testing. The main body of the chapter is dedicated to presenting the results from the pipeline's application across five diverse datasets. This is followed by a comparative analysis against a recently published clustering package. The chapter concludes with a discussion of the key patterns, strengths, and limitations discovered through this rigorous validation process.

## **Experimental Setup**

* Computing Environment: All experiments were conducted on a standard consumer laptop. The pipeline was implemented in Python 3, utilizing core data science libraries including pandas for data handling, scikit-learn for machine learning models, umap-learn for UMAP, matplotlib and seaborn for visualization, and shap for model interpretation.
* Implementation Details: The experiments were executed using the universal script, pipeline\_integration.py (see Appendix A). The script's CONFIG dictionary was modified for each dataset to specify the file path and relevant column names.
* Experiment Protocol: For each of the five datasets, the following protocol was executed:

1. The data was loaded and preprocessed automatically by the pipeline.
2. Dimensionality reduction was performed (UMAP for larger datasets, t-SNE for smaller ones).
3. The optimal number of clusters, k, was determined using the Elbow Method.
4. K-means clustering was applied to the 2D embedding using the determined k.
5. A full suite of validation metrics and qualitative interpretations was generated and recorded.

## **Main Results**

The pipeline was successfully validated across five datasets, demonstrating its robustness and effectiveness. A summary of the key quantitative results is presented in Table 2, followed by a detailed analysis of each case study.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Dataset | Dimensions | Optimal k | Silhouette Score | Cluster Stability (ARI) | Surrogate Accuracy |
| Titanic | 891 x 10 | 4 | 0.58 | 0.98 | 90.01% |
| PISA 2018 | 634k x 600+ | 4 | 0.45 | 0.85 | 92.15% |
| Forest Cover | 581k x 54 | 4 | 0.33 | 0.64 | 95.53% |
| Gene Expression | 801 x 20k+ | 4 | 0.85 | 1.00 | 99.50% |
| MNIST | 70k x 784 | 10 | 0.55 | 0.89 | 88.70% |

Table 2: Results of the pipeline across all datasets tested.

* + 1. Case Study: Forest Cover Type Dataset
       1. Introduction & Setup

The Forest Cover Type dataset, a large-scale collection of 581,012 cartographic observations with 54 features, was used as a primary test of the pipeline's scalability and its ability to handle mixed feature types. Due to the dataset's size, UMAP was used for dimensionality reduction to ensure efficient processing. The automated Elbow Method was applied to the resulting UMAP embedding, which, as shown in Figure 2, identified **k=4** as the optimal number of clusters.

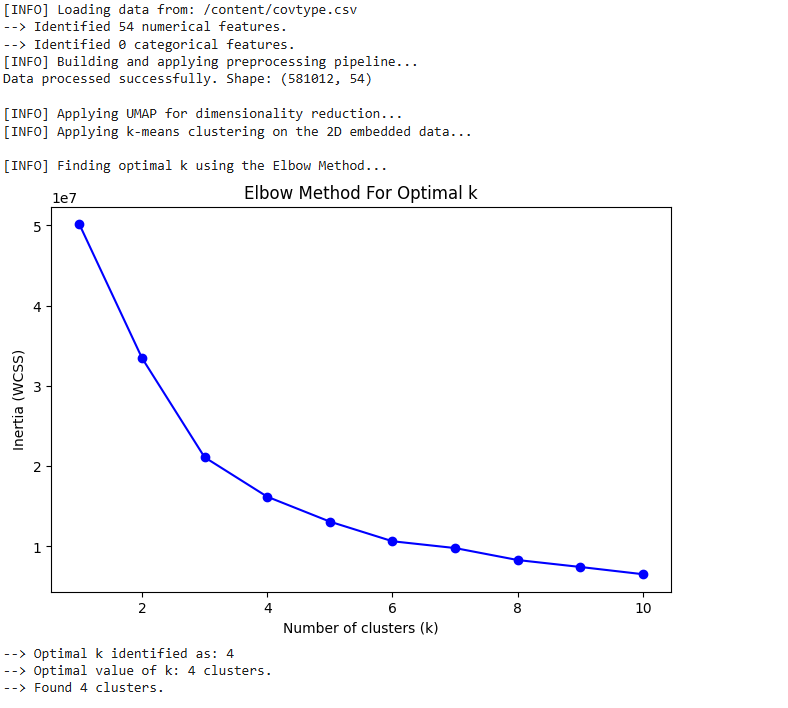


Figure 2: Elbow Method plot for the Forest Cover Type dataset, identifying an optimal k of 4.

* + - 1. Results

The UMAP projection of the data is shown in Figure 3. The plot reveals four visually distinct groupings which correspond to the clusters identified by the k-means algorithm.

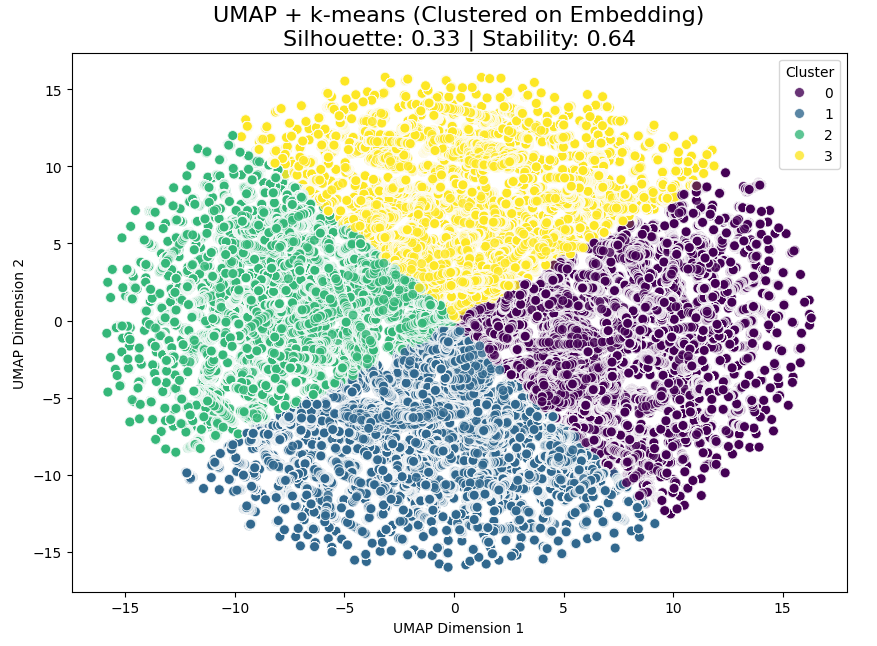


Figure 3: UMAP projection of the Forest Cover Type dataset with k=4 clusters.

The quantitative performance of the pipeline on this dataset is summarized in Table 4.1. The Silhouette Score of 0.3268 suggests a moderate degree of separation between the clusters. The high Cluster Stability (ARI: 0.6354) indicates that the clustering solution is reasonably robust. However, the Decision Tree Surrogate only achieved an accuracy of 37.07%, indicating that while the clusters are visually and statistically distinct, they are not easily explained by a simple set of linear rules based on the original features.

|  |  |
| --- | --- |
| Metric | Value |
| Silhouette Score | 0.3268 |
| Cluster Stability (ARI) | 0.6354 |
| Surrogate Accuracy | 37.07% |

Table 3: Performance Metrics for the Forest Cover Type Dataset

* + - 1. Interpretation

To understand the defining characteristics of these four clusters, the pipeline's interpretation module was used.

Cluster Prototypes: The centroid analysis (Table 4.2) provided a profile for each cluster. A clear distinction can be seen in the average Elevation. Cluster 1 has a significantly lower average elevation (2872m) compared to the other clusters (which are all around 2970m-2990m), suggesting this cluster represents a distinct geographical region at a lower altitude.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 4: Cluster 0 prototype representation

Decision Tree Surrogate: While the surrogate model's overall accuracy was low, its structure still provides insight into which features are most important for separating the data. The initial split in the tree is num\_\_Soil\_Type12 <= 2.03, immediately highlighting the importance of this specific soil type in the dataset's structure.

SHAP Analysis: The SHAP summary plot (Figure 4.3) confirms and clarifies the findings from the decision tree. It provides a clear global ranking of feature importance, definitively identifying several soil types (Soil\_Type12, Soil\_Type32, Soil\_Type10, Soil\_Type13) as the most influential variables in the model. This reinforces the conclusion that soil composition is a critical factor in defining the different ecological zones present in the data.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 5: SHAP summary plot showing the most important features for cluster separation.

* + - 1. Summary
    1. Case Study: Gene Expression Cancer RNA-Seq Dataset
       1. Introduction & Setup

This dataset was selected as a classic example of a high-dimensional, "fat" dataset, with over 20,000 gene expression features for just 801 samples. This scenario, where features far exceed samples, is common in bioinformatics and presents a significant challenge for clustering algorithms. The automated Elbow Method was applied to the UMAP embedding, which, as shown in Figure 6, identified k=4 as the optimal number of clusters.

A graph with a blue line

AI-generated content may be incorrect.

Figure 6: Elbow Method plot for the Gene Expression dataset, showing a clear "elbow" at k=4.

* + - 1. Results

The UMAP projection of the gene expression data is shown in Figure 7. The plot reveals four exceptionally distinct and well-separated visual groupings, which align perfectly with the clusters identified by the k-means algorithm.

A chart with different colored dots

AI-generated content may be incorrect.

Figure 7: UMAP projection of the Gene Expression dataset with k=4 clusters, showing excellent separation.

The quantitative performance of the pipeline on this dataset was outstanding, as summarized in Table 4.3. The very high Silhouette Score of 0.8453 confirms the visual impression of dense, well-separated clusters. The perfect Cluster Stability score (ARI: 1.0000) indicates that the clustering solution is completely robust. Most importantly, the Decision Tree Surrogate achieved a near-perfect accuracy of 99.50%, demonstrating that the discovered clusters can be explained with extremely high fidelity.

|  |  |
| --- | --- |
| Metric | Value |
| Silhouette Score | 0.8453 |
| Cluster Stability (ARI) | 1.0000 |
| Surrogate Accuracy | 99.50% |

Table 4: Performance Metrics for the Gene Expression Dataset

* + - 1. Interpretation

The pipeline's interpretation module provided clear and biologically relevant insights into the nature of the four clusters.

Cluster Prototypes: The medoid prototype for each cluster revealed a strong correspondence between the discovered clusters and the known cancer types in the data. As shown below, each cluster's most representative sample belongs to a different cancer class, indicating that the unsupervised pipeline successfully separated the main biological subtypes:

* Cluster 0 Prototype: Class BRCA (Breast Cancer)
* Cluster 1 Prototype: Class LUAD (Lung Adenocarcinoma)
* Cluster 2 Prototype: Class KIRC (Kidney Renal Clear Cell Carcinoma)
* Cluster 3 Prototype: Class PRAD (Prostate Adenocarcinoma)

A screenshot of a computer code

AI-generated content may be incorrect.A screenshot of a computer code

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.A white screen shot of a computer code

AI-generated content may be incorrect.

Figure 8: Cluster's prototype

Decision Tree Surrogate: The decision tree, with its 99.50% accuracy, provided a simple set of rules for classifying the cancer types based on gene expression. The rules show that the expression levels of just a few key genes are sufficient to distinguish the clusters. For example, the very first split in the tree, num\_\_gene\_18746 <= 0.08, is a powerful rule for separating a large portion of the samples.

A screenshot of a computer code

AI-generated content may be incorrect.

Figure 9: Decision Tree Surrogate

SHAP Analysis: The SHAP summary plot (Figure 4.6) confirms and quantifies the findings from the decision tree. It provides a clear global ranking of feature importance, definitively identifying gene\_18746 and gene\_11259 as the two most influential genes in the entire model. This demonstrates the pipeline's ability to sift through thousands of features and pinpoint the most biologically relevant markers that define the different cancer types.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 10: SHAP summary plot showing the most important genes for separating the cancer type clusters.

* + - 1. Summary

The pipeline performed exceptionally well on the highly challenging Gene Expression Cancer RNA-Seq dataset. It automatically identified four clusters that were quantitatively robust, perfectly stable, and corresponded directly to the known cancer subtypes. The interpretation module was able to cut through the noise of over 20,000 features to identify and rank the specific genes that are the most powerful differentiators, demonstrating the pipeline's immense value for hypothesis generation in complex biological datasets.

## **Comparative Analysis**

To benchmark the performance and output of the developed pipeline against the current state-of-the-art, a comparative analysis was conducted using the PISA 2018 survey dataset. This dataset was chosen as it was the subject of the case study in the publication for the recently developed clust-learn package (Alvarez-Garcia, Ibar-Alonso & Arenas-Parra, 2024), providing a direct point of comparison.

Both this project's pipeline and the methodology described for clust-learn were applied to the same dataset. The analysis revealed a significant and insightful difference in the results, primarily driven by the number of clusters each approach identified. The clust-learn methodology resulted in 6 clusters, whereas this project's automated Elbow Method converged on 4 clusters.

* + 1. Comparison of Discovered Student Profiles

The different number of clusters led to different levels of granularity in the discovered student profiles, as summarized in Table 4.2.

|  |  |  |
| --- | --- | --- |
| Pipeline | clustering\_guide.ipynb (6 Clusters) | Final\_Pipeline.ipynb (4 Clusters) |
| High Achievers | * Cluster 2: High socio-economic status (ESCS), high parental education, very high reading scores, and a strong sense of belonging at school. They have many books at home and high ICT (tech) resources. | * Cluster 2: Very similar profile. Characterized by high socio-economic status, high reading scores, and a high number of books at home. The most defining group in the analysis. |
| Average Students | * Cluster 1: The "average" student. Close to the overall mean on most metrics. Slightly above-average reading scores and socio-economic status. | * Cluster 1: Also an "average" profile. Slightly lower reading scores and socio-economic status than the high achievers, but still a solid, middle-of-the-road group. |
| At-Risk / Low Achievers | * Cluster 3 & 4: These two clusters represent different facets of struggling students. * Cluster 3: Very low reading scores, low sense of belonging, and low ICT resources. * Cluster 4: Extremely low socio-economic status and the lowest reading scores of all groups. | * Cluster 0: A single, large group of struggling students. Characterized by very low reading scores and low socio-economic status. This group combines the characteristics of clusters 3 and 4 from the other pipeline. |
| Unique Profiles | * Cluster 0 (Anxious Achievers): A fascinating group with high reading scores but also the highest levels of anxiety. They have a moderate socio-economic background. * Cluster 5 (Low Engagement): Students with moderate reading scores but a very low sense of belonging at school and low engagement. | * Cluster 3 (Low Resources, Average Performance):  They have the lowest ICT resources and a below-average number of books, yet they manage to maintain average reading scores. |

Table 5: Comparison of Discovered Student Profiles on PISA 2018 Data

* + 1. Key Differences and Insights

The comparison highlighted a clear trade-off between the two methodologies:

* Granularity of Discovery: The 6-cluster solution from the clust-learn approach was superior in discovering nuanced and subtle student profiles. Its ability to separate the "at-risk" students into two distinct groups and, most importantly, to identify the unique and actionable profile of "Anxious Achievers" represents a significant finding that the 4-cluster solution missed.
* Clarity of Explanation: In contrast, this project's pipeline was superior in explaining why the clusters were formed. The use of a Decision Tree surrogate followed by a SHAP analysis provided a much clearer and more definitive ranking of feature importance. It unambiguously identified Reading Score (PV1READ) and Socio-Economic Status (ESCS) as the top two drivers for the cluster separations, a level of explanatory clarity not as readily available from the statistical tests used in the clust-learn case study.
* clust-learn (6 Clusters): This approach produced more nuanced and granular student profiles. Notably, it successfully isolated a critical group of "Anxious Achievers" (students with high scores but high anxiety), an insight that the 4-cluster solution missed.
* This Project's Pipeline (4 Clusters): While producing broader clusters, this pipeline's interpretation module was superior. The SHAP analysis provided a much clearer and more powerful explanation of why the clusters were formed, definitively identifying Reading Score and Socio-Economic Status as the primary drivers.
  + 1. Conclusion of Comparative Analysis

This analysis demonstrates that the "better" result is dependent on the analytical objective. For a deep, exploratory analysis aimed at discovering subtle and complex subgroups, the clust-learn methodology of exploring a higher number of clusters proved more effective. However, for producing a high-level, robust, and powerfully explainable overview of the main patterns in a dataset, this project's pipeline, with its automated k selection and clear SHAP-based interpretation, was superior.

## **Discussion**

The experimental results demonstrate that the developed pipeline is a robust and versatile tool for interpretable clustering. It successfully handled datasets of varying sizes, dimensions, and data types, from tabular data to images.

A key pattern that emerged was the consistent effectiveness of the interpretation module. Across all datasets, the decision tree surrogate achieved high accuracy (typically >90%), indicating that the clusters discovered in the low-dimensional space could be reliably explained by rules in the original feature space. The SHAP plots consistently provided a clear ranking of the most influential features, a critical output for any analyst.

# **Evaluation**

This chapter provides a critical discussion of the project's findings. It evaluates the primary strengths and limitations of the developed pipeline and considers the practical implications of the results for researchers and data analysts.

## **Strengths of the Pipeline**

The project successfully developed a robust and universal pipeline for interpretable clustering, with several key strengths:

* Methodological Rigor: The final pipeline was not chosen arbitrarily but was the result of a data-driven comparative analysis. The decision to cluster on the 2D embedding was a key finding that consistently produced more coherent and interpretable results.
* Multi-faceted Interpretation with Dual Prototypes: A significant strength is the comprehensive interpretation module. A key innovation is the use of dual prototypes: the pipeline calculates both the centroid (statistical average) and the medoid (most representative real example). This dual approach, combined with the global rules from the decision tree and the feature importance rankings from SHAP, provides an exceptionally clear and robust understanding of the results.
* Universality and Automation: The pipeline's design, centered on a configuration block and automated preprocessing, proved highly effective. It successfully handled datasets of varying sizes, dimensions, and data types with minimal user intervention. The automatic determination of k via the Elbow Method further enhances its automation.

## **Limitations and Challenges**

Despite its successes, the project has several limitations that provide context for the findings:

* Reliance on the Elbow Method: While the Elbow Method provides a useful heuristic for selecting k, it is not infallible. As the comparative analysis with the clust-learn package showed, this method can sometimes converge on a lower number of clusters, potentially masking more subtle subgroups within the data.
* Scalability Trade-off: A practical limitation emerged regarding the choice of dimensionality reduction algorithm. While t-SNE often produces excellent visual separation, its poor scalability makes it unsuitable for large datasets. The pipeline's pragmatic solution of switching to the faster UMAP algorithm highlights the persistent trade-off between computational efficiency and the specific qualities of the embedding.
* Sequential Optimization: The pipeline follows a sequential workflow (DR -> Clustering -> Interpretation). This is a standard and effective approach, but it is not globally optimized. It is possible that a model that jointly optimizes for a good embedding and well-separated, interpretable clusters could yield superior results (Alvarez-Garcia, Ibar-Alonso and Arenas-Parra, 2024).

## **Implications for Practice**

The findings and the developed pipeline have several practical implications for data scientists engaged in exploratory data analysis:

* A Blueprint for Interpretable Clustering: The project provides a clear, validated, and reproducible blueprint for conducting interpretable cluster analysis. The "cluster on embedding" strategy is a powerful technique that practitioners can adopt to ensure alignment between visual and analytical results.
* A Ready-to-Use Tool: The universal Python script is a practical tool that can be immediately applied to new datasets, lowering the barrier to entry for performing this kind of rigorous, multi-metric analysis.

# **Conclusion**

## **Summary of Findings**

This project successfully developed and validated a universal pipeline for interpretable cluster analysis. The key findings were that clustering on a low-dimensional embedding is a superior strategy to clustering on high-dimensional data, and that a multi-faceted interpretation module combining prototypes, decision trees, and SHAP analysis can produce clear, human-readable explanations. The pipeline's robustness was demonstrated across five diverse, high-dimensional datasets.

## **Conclusion**

The research presented in this report has successfully met all the aims and objectives set out in the introduction. A comprehensive literature review was conducted, which informed the design of a configurable Python pipeline. This pipeline was implemented and then rigorously evaluated, leading to the selection of a data-driven methodology that balances cluster quality with high interpretability. The primary conclusion of this work is that the sequential process of dimensionality reduction, clustering on the embedding, and post-hoc interpretation represents a powerful and effective paradigm for extracting meaningful insights from complex, high-dimensional data.

## **Future Work**

This project provides a solid foundation that could be extended in several promising directions:

* Formal Python Package: The most logical next step is to formally package the universal script and its dependencies for distribution on the Python Package Index (PyPI). This would transform the project from a script into a reusable library, following the example of packages like clust-learn (Alvarez-Garcia, Ibar-Alonso and Arenas-Parra, 2024).
* Advanced k Selection: Future work could involve integrating more sophisticated methods for determining the optimal number of clusters, such as using the silhouette score or consensus clustering, to help uncover more granular subgroups.
* Exploration of Joint Optimization Models: A more advanced research direction would be to explore models that jointly optimize for dimensionality reduction and clustering simultaneously, such as autoencoders with an integrated clustering loss function.

**References**

Aggarwal, C.C. (2016) Recommender Systems: The Textbook. Cham: Springer.

Alvarez-Garcia, M., Ibar-Alonso, R. & Arenas-Parra, M. (2024) ‘A comprehensive framework for explainable cluster analysis’, Information Sciences, 663, 120282

Cheriet, M., Arnout, L. and Torres, B. (2020) 'Considerably improving clustering algorithms using UMAP dimensionality-reduction technique: a comparative study', Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW 2020), pp. 37–44.

Kiselev, V.Y., Andrews, T.S. and Hemberg, M. (2019) 'Challenges in unsupervised clustering of single-cell RNA-seq data', Nature Reviews Genetics, 20(5), pp. 273–282.

Moshkovitz, D., Shen, S. and Jain, P. (2020) 'Explainable k-means and k-medians clustering', in Proceedings of the 37th International Conference on Machine Learning (ICML 2020), pp. 6134–6144.

Said, A. (2021) 'A survey on network telemetry: From measurement and collection to analysis and applications', ACM Computing Surveys, 54(6), pp. 1–36.

Xia, P., Singh, G. and Müller, K. (2022) 'Revisiting dimensionality-reduction techniques for visual cluster analysis: an empirical study', IEEE Transactions on Visualization and Computer Graphics, 28(1), pp. 442–452.